

# TGVizTab: An Ontology Visualisation Extension for Protégé

Harith Alani

IAM Group, ECS Dept.  
University of Southampton  
Southampton, Hampshire, UK  
email: h.alani@ecs.soton.ac.uk

## ABSTRACT

Ontologies are gaining a lot of interest and many are being developed to provide a variety of knowledge services. There is an increasing need for tools to graphically and interactively visualise such modelling structures to enhance their clarification, verification and analysis. Protégé 2000 is one of the most popular ontology modelling tools currently available. This paper introduces TGVizTab; a new Protégé plugin based on TouchGraph technology to graphically visualise Protégé's ontologies.

## Categories and Subject Descriptors

E.1 Data Structures – *graphs and networks*. I.2.4 Knowledge Representation Formalisms and Methods – *semantic networks*.

## Keywords

Ontology visualisation, network graphs, ontology navigation.

## INTRODUCTION

An ontology is a conceptualisation of a domain into machine readable format [6]. Ontologies are becoming increasingly popular modelling schemas for knowledge management services and applications. Focus on developing tools to graphically visualise ontologies is rising to aid their assessment and analysis. Graph visualisation helps to browse and comprehend the structure of ontologies, analyse their network connections, observe changes, etc.

Protégé-2000 [13] is a very popular knowledge-modelling tool developed at Stanford University. Ontologies and knowledge-bases (KBs) can be edited interactively within Protégé and accessed with a graphical user interface and Java API. Protégé can be extended with pluggable components to add new functionalities and services. There exists an increasing number of plugins offering a variety of additional features, such as extra ontology management tools, multimedia support, querying and reasoning engines, problem solving methods, etc.

This paper describes TGVizTab; an ontology visualisation plugin for Protégé based on TouchGraph<sup>1</sup>. The plugin aims

at helping Protégé users to visualise and browse their ontologies and KBs via interactive dynamic graphs.

Many tools exist for visualising ontological networks. Nevertheless, some of these tools lack support for visualising certain ontology characteristics or are designed for a specific range of queries. The motivation behind TGVizTab is to develop a lightweight ontology visualisation tool that caters for common ontology features and meets some of the special requirements for visualising such network structures. The following sections introduce several of these requirements, and demonstrate how some of them are handled by TGVizTab.

## ONTOLOGY VISUALISATION

A number of visualisation techniques have been described over the years, such as spanning tree layouts, tree-maps [9], fisheye views [4], hyperbolic [10] and 3D hyperbolic layouts [12], aiming to help comprehend and analyse complex information structures. Preference of visualisation models vary according to users needs and query context [5]. It is also dependant on the type and extent of the visualised network. Using a combination of integrated visualisations of various types has shown to be sometimes beneficial [15][16].

Complex networks of multi dimensional hierarchies and arbitrary relations are becoming common characteristics of current ontologies. Tools that discriminate against some of these features, for example by supporting spanning trees or hierarchical relations only, might not be appropriate for comprehensive ontology visualisation.

Ontologies, together with their KBs, could grow into very large information networks, especially if aimed at providing scalable services for the Semantic Web [1]. Visualising large networks has always been challenging [20]. [8] surveyed a wide range of visualisation techniques and concluded that all existing algorithms have a size limit after which they cannot cope. [14] and [8] stressed the importance of reducing the visualised graphs into smaller sized *sub graphs* that users can browse to visualise other parts of the network.

Ontologies are semantically rich by definition. Ontology visualisers should therefore turn some of these semantics more explicit [20]. Spring-layout algorithms [2] are example techniques that display semantically similar nodes closer to each other. Such layouts could help users to

<sup>1</sup> <http://www.touchgraph.com/>

quickly recognise dense areas and interrelated objects in their ontologies and KBs.

## TOUCHGRAPH

TouchGraph is an open source Java environment for the creation and navigation of interactive network graphs. TouchGraph offers several desirable features for visualising networks, such as high level of interactivity, fast rendering, pan and zoom capability, locality control, etc. Such characteristics have been shown to be vital for visualising large information networks [8] [20].

TouchGraph applies a spring-layout technique, where nodes repel each other while edges (connections) attract them. This results in the placement of semantically similar nodes closer to each other.

TouchGraph is being used in many graphical applications, such as GoogleBrowser<sup>2</sup> which displays web page linkages (Figure 1), AmazonBrowser<sup>3</sup> for showing graphs of similar items, PubMed<sup>4</sup> for visualising graphs of related documents in a medical library, Ol-Modeler (11) for editing and managing ontologies, Semaview's<sup>5</sup> applet to visualise social networks, and Planetarium<sup>6</sup>; an interactive graph-based web navigator.

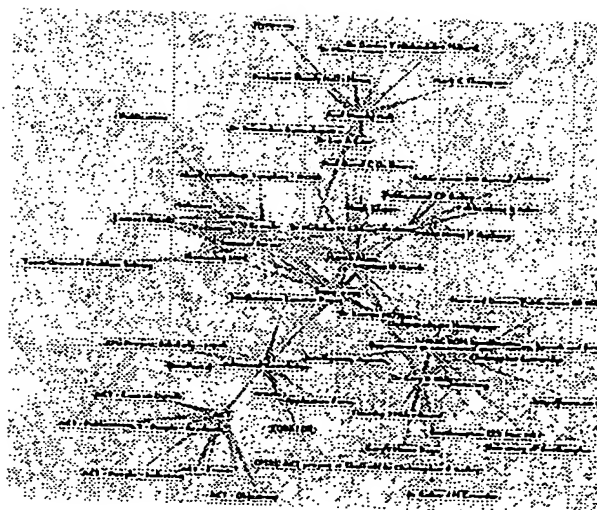


Figure 1. GoogleBrowser. Uses TouchGraph to display networks of web pages

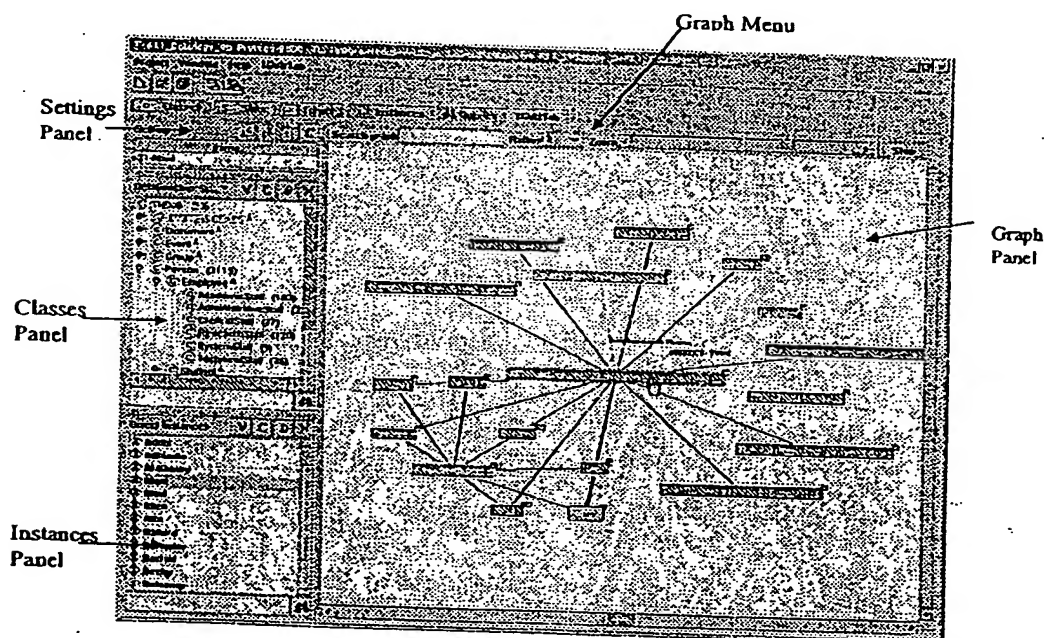


Figure 2. The user interface of the Protégé TGVizTab plugin.

- <sup>2</sup> <http://www.touchgraph.com/TGGoogleBrowser.html>
- <sup>3</sup> <http://www.touchgraph.com/TGAmazonBrowser.html>
- <sup>4</sup> <http://www.touchgraph.com/TGPubMedBrowser.html>
- <sup>5</sup> <http://www.semaview.com/>
- <sup>6</sup> <http://www.touchgraph.com/Planetarium.html>

## TGVIZTAB PLUGIN

TGVizTab<sup>7</sup> (TouchGraph Visualisation Tab) integrates TouchGraph technology with Protégé to create a new plugin for visualising ontologies and KBs as directed network graphs of classes, instances, and relations.

TGVizTab is generic, dynamic (graphs created on the fly), and customisable. TouchGraph was modified in TGVizTab to read directly from Protégé's Java API. It was also extended to cater for specific ontology visualisation requirements, such as handling different types of relations and edge labelling.

TGVizTab graphs can be saved in XML and viewed with other TouchGraph applications.

## User Interface

Figure 2 shows the user interface of TGVizTab. The ontology schema is displayed on the left (Classes Panel) using the tree-list layout provided by Protégé. When a class is selected, the list of its direct instances will be displayed in the Instances Panel. To generate a graph, a class or an instance has to be selected to act as the graph's focal node. This can be done by selecting the class or instance of interest from the related panels and adding it to the Settings Panel (Figure 3) using the relevant buttons (+ to add a class, and + to add an instance). The Settings Panel contains a set of menu tabs for colour customisation, relation visibility, node expansion thresholds, etc.

When the C button is clicked, the corresponding graph will be created and displayed in the Graph Panel. The Graph Menu (Figure 2) offers several graph control features, such as zoom, node search, and locality control.

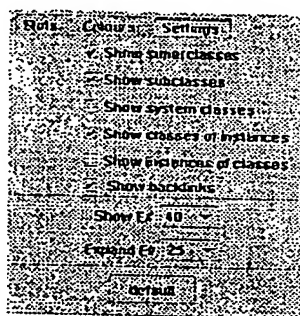


Figure 3. TGVizTab's Settings Panel.

## Nodes and Edges

TGVizTab displays classes as well as instances. These are represented as nodes in the graph, and can be displayed in different colours to be easily distinguishable. Ontology relations (slots) are represented as graph edges (links between the nodes).

<sup>7</sup> <http://www.ccs.soton.ac.uk/~hs/TGVizTab>

When visually analysing an ontology, it is sometimes necessary to clearly identify the different type of connections in a graph. Furthermore, users may wish to study a graph based on specific ontology relations, while hiding all other relations. To this end, TGVizTab users are given full control over the colour and visibility of each ontology relation, which can be modified in the Settings Panel (Figure 3).

Graph edges are labelled after the ontology relations they represent. To avoid clutter, edge labels are only displayed when hovering the mouse over the edges, or over their originating nodes (Figure 2).

Right clicking a node brings up the Node Menu (Figure 4). This menu contains options for hiding, expanding, and collapsing individual nodes, and for viewing their Protégé description forms.

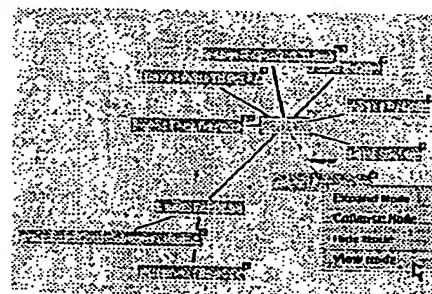


Figure 4. Node Menu.

## Incremental Graph Navigation

TGVizTab allows Protégé users to gradually visualise their ontologies by interactively navigating connected sub-graphs. The Radius value in the Graph Menu (Figure 2) indicates the locality of the displayed graph, which is the maximum network depth from the focal node. The radius is used to limit the graph to smaller, manageable sizes.

When visualizing sub graphs, it becomes necessary to be able to navigate to other parts of the graph with minimum disturbance and in a timely fashion. In TGVizTab, double clicking a graph node causes the graph to recreate itself around the clicked node. This transition takes place dynamically and animatedly to minimise disruption and ensure steady navigation.

To reduce screen clutter, two thresholds are introduced by TouchGraph and adopted in TGVizTab. These are the *Show* and *Expand* thresholds (Figure 3) which represent the maximum number of edges a node is allowed to have to stay visible and expanded respectively. These thresholds provide users with control over the visibility and expansion of heavily connected nodes. Initial observations showed these confinements to be useful for reducing the visual complexity of dense graphs.

When a node is partially expanded, a small box (Figure 4) is displayed next to the node to indicate the number of hidden edges. This TouchGraph feature enables users to easily

recognise the nodes that could be expanded further. Increasing locality or expanding a node from the Node Menu can display some of these hidden edges.

### Pan & Zoom

[8] describe two useful types of zooming, *Geometric*, and *Semantic*. Currently, zooming in TGVizTab is only geometric, where edges are stretched or shrank when zooming in or out respectively. Some tools, such as Netscan [21] and Jambalaya [19] provide semantic zooming to adjust the level of visible detail.

Truncating labels can sometimes cause loss of information [5]. TGVizTab currently applies no restriction on the maximum displayable length of node and edge labels. However, clutter may increase dramatically if labels are too long. Semantic zooming or adjustable label size limits could leverage this problem.

### Orientation

The familiar tree-list layouts used in Protégé for the display of classes and instances are integrated with TGVizTab to help users easily locate objects of interest. However, implicit relationships between multiple views are often difficult for users to envisage [18]. TGVizTab helps to ease this problem by *hyperlinking* each graph node with its corresponding Protégé class or instance. When a graph node is clicked, the class or instance it represents will be listed and highlighted in the panels on the left (Figure 2). This helps users to conceptually map the graph to the ontology by allowing them to browse directly from any node in the graph to its representation in the ontology.

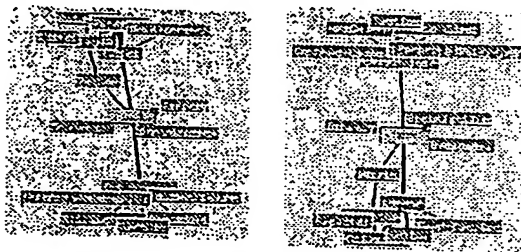


Figure 5. Two graphs of the same network

Predictability of graphs refers to how consistent the drawing of a graph is, i.e. how the graph changes each time it is drawn [14][8]. Rotation and projection of graphs created by TGVizTab are not predictable as they may change slightly each time they are drawn (Figure 5). Unpredictability may cause high interaction difficulties, but dealing with small or sparse graphs may ease this problem [8].

### RELATED WORK

Currently there exists two other graphical visualisation plugins for Protégé; OntoViz<sup>8</sup> (Figure 6), which is based on

<sup>8</sup> <http://protege.stanford.edu/plugins/ontoviz/ontoviz.html>

AT&T's GraphViz software, and Jambalaya [19] (Figure 7), which is based on ShriMP [18]. OntoViz provides customisable graphical visualisations of Protégé's ontologies. Unlike TGVizTab, OntoViz permits visualising several disconnected graphs at once. Small sized graphs of OntoViz are suitable for presentation purposes, as they tend to be of good clarity with none overlapping nodes. However, OntoViz graphs are static and non-interactive, yielding them less suitable for browsing or visualising large networks.

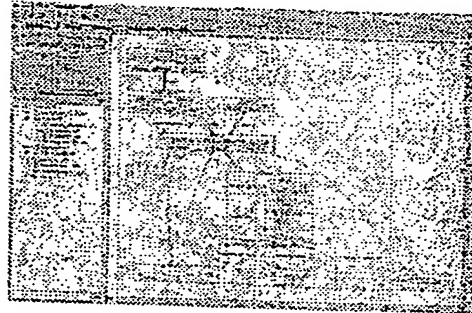


Figure 6. OntoViz showing a simple ontology in Protégé

ShriMP is an information visualisation technique for navigating abstracted structures of complex information networks with animated semantic zoom. Jambalaya is the product of merging ShriMP with Protégé. It offers interchangeable views of nested graphs for interactive ontology navigation and knowledge acquisition. Unlike the other two Protégé visualisation extensions (i.e. OntoViz and TGVizTab), Jambalaya offers a range of interconnected layouts to meet different user preferences and query requirements. Jambalaya is concerned with providing a fully-fledged knowledge visualisation system, while TGVizTab is meant to be a lightweight visualisation solution.

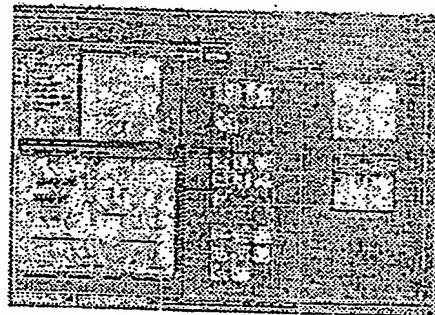


Figure 7. A Jambalaya display of an ontology in Protégé  
Hyperbolic techniques, such as those used in NichWorks [20] (Figure 8), H3Viewer [12], and Walrus<sup>9</sup> (Figure 9),

<sup>9</sup> <http://www.caida.org/tools/visualization/walrus/>

demonstrated a capability of displaying thousands of nodes and edges on single screens. However, such graphs are often difficult to comprehend and analyse due to their sheer size, but can be useful for abstract overviews of whole networked structures [8].



Figure 8. NichWorks, showing the web space of a home page.

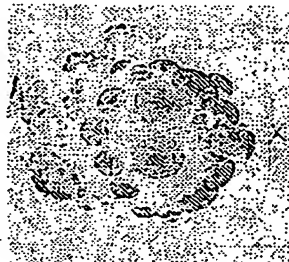


Figure 9. A Walrus view of a CVS directory with nearly 18.5k nodes and edges.

Inxight and OntoRama [3] (Figure 10) are two example viewers with 2D hyperbolic layouts. OntoRama visualises ontologies expressed in RDF. Both Inxight and OntoRama are limited to spanning tree graphs, which is not suitable for visualising multiple hierarchies and relation types.

Spring-layout algorithms similar to those of TouchGraph are used in Inxight, OntoRama [3], and NetScan [21] to position nodes with relation to their connectivity pattern. Some tools, such as NetScan and NichWorks [20], allow edge types to be associated with user-defined weights to influence their lengths.

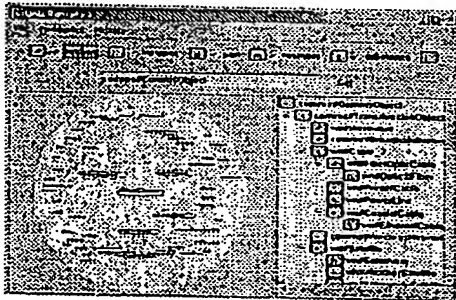


Figure 10. OntoRama visualisation of a communications' ontology

Grouping nodes into clusters according to certain semantics is one approach to reduce the visual complexity of large graphs [17]. Spectacle [7] (Figure 11) visualises ontologies as taxonomic clusters. Clusters in Spectacle represent groups of instances of individual classes or multiple classes. Spectacle displays class hierarchical relations only, while hiding any relations at the instance level.

A number of tools have been developed to provide incremental navigation of web pages, such as GoogleBrowser

(Figure 1) and WebBrain<sup>10</sup>. GoogleBrowser visualises hyperlinked pages using TouchGraph. WebBrain navigation is confined to a simple taxonomy of topics, and is strictly limited to the immediate neighbourhood of web pages.

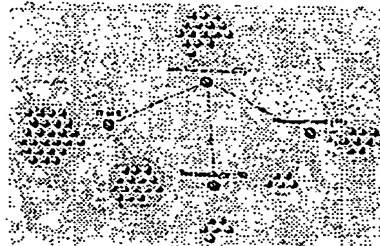


Figure 11. Jobs vacancy clusters in Spectacle.

OI-Modeler [11] (figure 12) is a tool for the creation and maintenance of ontologies with a graphical layout based on TouchGraph. The focus of OI-Modeler is on ontology change management, rather than generic purpose visualization as in TGVizTab.



Figure 12. OI-Modeler for editing ontologies.

## FUTURE WORK

User feedback showed a high interest in the plugin, and many of their suggestions and requests will be taken on board in the following releases.

Experiments are planned to observe how the plugin is used and for what type of queries. This will help discover any further requirements to improve the functionality and use of TGVizTab.

## CONCLUSIONS

Visualisation is becoming an important part in the development life-cycle of ontologies. This paper described TGVizTab; a new Protégé visualisation plugin based on TouchGraph. Protégé 2000 is one of the most popular environments for building ontologies.

The aim of TGVizTab is to provide a scalable, lightweight, general-purpose visualisation of ontologies and KBs in the form of directed network graphs that can be navigated in-

<sup>10</sup> [http://www.webbrain.com/html/default\\_wbr.html](http://www.webbrain.com/html/default_wbr.html)

crementally and interactively. The plugin caters for several ontology visualisation needs, such as the handling of both classes and instances, incremental exploration of large graphs, explicit semantics, etc.

### ACKNOWLEDGEMENT

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UKEPSRC under grant number GR/N15764/01. The AKT IRC comprises the Uni. of Aberdeen, Edinburgh, Sheffield, Southampton and the Open Uni. Thanks also to TouchGraph LLC, and to Alex Shapiro in particular for the software and support. The Protégé resource is supported, in part, by grant P41 LM007885 from the National Library of Medicine.

### REFERENCES

- [1] Berners-Lee, T., Hendler, J., Lassila, O.: *The Semantic Web*. Scientific American, 2001.
- [2] Eades, P.: A Heuristic for Graph Drawing. *Congressus Numerantium*, 42:149-160, 1984.
- [3] Eklund, P. W., Roberts, N., Green, S. P.: *OntoRama: Browsing an RDF Ontology using a Hyperbolic-like Browser*. First Int. Symp. on CyberWorlds (CW2002), IEEE press, pp 405-411, 2000.
- [4] Furnas, G. W.: The FISHEYE view: A new look at structured files. *Proc. of the Conf. on Human Factors in Computing Systems ACM*, pp 16-23, 1986.
- [5] Graham, M., Kennedy, J., Benyon, D.: Towards a Methodology for Developing Visualizations. *Int. J. of Human-Computer Studies* 53(5): 789-807, 2000.
- [6] Guarino, N., Giaretta, P.: Ontologies and Knowledge bases: towards a terminological clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS, 25-32, 1995.
- [7] Harmelen, F. van, Brockstra, J., Fluit, C., Horst, H., Kampman, A., van der Meer, J., Sabou, M.: *Ontology-based Information Visualisation*. Workshop on Visualisation of the Semantic Web, London, 2001.
- [8] Herman, I., Melancon, G., Marshall, M.S.: Graph Visualization and Navigation in Information Visualisation: a Survey. *IEEE Transactions on Visualisation and Computer Graphics* 6(1): 24-43, 2000.
- [9] Johnson, B., Shneiderman, B.: Treemaps: a space-filling approach to the visualisation of hierarchical information structures. *Proc. 2<sup>nd</sup> Int. IEEE Visualization Conference*, San Diego, 1991.
- [10] Lamping, J., Rao, R., Pirolli, P.: A focus + context Technique based on Hyperbolic Geometry for Visualizing Large Hierarchies. *ACM Conference on Human Factors in Computing Systems (CHI'95)*, New York, ACM Press, pp 404-408, 1995.
- [11] Maedche, A., Motik, B., Stojanovic, L., Studer, R., Volz, R.: An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies. *Proc. 12<sup>th</sup> Int. World Wide Web Conference (WWW'03)*, Budapest, Hungary, ACM, 439-448, 2003.
- [12] Munzner, T.: H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. *Proc. of the IEEE Symp. on Information Visualisation*, Phoenix, USA, 1997.
- [13] Muscu, M.A., Ferguson, R.W., Grosso, W.E., Noy, N.F., Grubezy, M.Y., Gennari, J.H.: Component-based support for building knowledge-acquisition systems. *Proc. Intelligent Information Processing (IIP 2000) Conf. Int. Federation for Processing (IFIP), World Computer Congress (WCC2000)*, Beijing, China, pp 18-22, 2000.
- [14] North, S. C.: Incremental layout in DynaDAG. *Proc. of the Symposium on Graph Drawing GD '95*, Springer-Verlag, pp 409-418, 1995.
- [15] North, C., Shneiderman, B.: Snap-together visualisation: can users construct and operate coordinated visualisations?. *Int. J. of Human-Computer Studies* 53: 715-739, 2000.
- [16] Ridsen, K., Czerwinski, M.P., Munzner, T., Cook, D.: An initial examination of ease of use for 2D and 3D information visualizations of web content. *Int. J. of Human-Computer Studies* 53: 695-714, 2000.
- [17] Roxborough, T., Sen, A.: Graph Clustering Using Multiway Ratio Cut. *Proceedings of Symposium on Graph Drawing (GD'97)*, Berlin, Germany, LNCS, vol. 1353, Springer-Verlag, pp 291-296, 1997.
- [18] Storey, M.A., Wong, K., Fracchia, F.D., Muller, H.A.: On Integrating Visualization Techniques for Effective Software Exploration. *Proc. IEEE Symp. on Information Visualization (InfoVis'97)*, Phoenix, Arizona, USA, 38-45, 1997.
- [19] Storey, M.A., Muscu, M., Silva, J., Best, C., Ernst, N., Ferguson, R., Noy, N.: Jambalaya: Interactive visualisation to enhance ontology authoring and knowledge acquisition in Protégé. *Workshop on Interactive Tools for Knowledge Capture, K-CAP-2001*, Victoria, B.C. Canada, 2001.
- [20] Wills, G. J.: NicheWorks - Interactive Visualisation of Very Large Graphs. *Proc. Graph Drawing '97*, Rome, Italy, Springer-Verlag, pp 403-414, 1997.
- [21] Xiong, R., Smith, M. A., Drucker, S. M.: Visualisation of Collaborative Information for End-Users, Microsoft MST-TR-98-52, 1998.